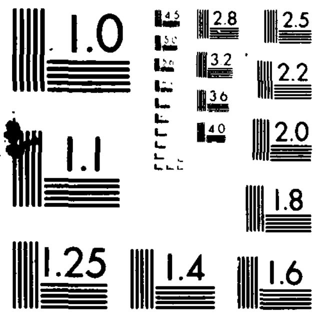
AD-A110 218 MITRE CORP BEDFORD MA F/G 17/2.1

AN ALGORITHM SCHEMA FOR MINIMIZING BROADCAST TRANSMISSIONS IN P--ETC(U)

DEC 81 B W FAM F19628-81-C-0001

UNCLASSIFIED MTR-8384 ESD-TR-81-388 NL

END
DATE
FILMED
2 82
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

LEVEL II

(2)

ESD-TR-81-388

MTR-8384

AD A110218

AN ALGORITHM SCHEMA FOR MINIMIZING BROADCAST TRANSMISSIONS IN PACKET RADIO NETWORKS

By

Bahaa W. Fam

DECEMBER 1981

Prepared for

DEPUTY FOR COMMUNICATIONS AND INFORMATION SYSTEMS
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
Hanscom Air Force Base, Massachusetts

Approved for public release;
distribution unlimited.

Project No. 8280

Prepared by

THE MITRE CORPORATION
Bedford, Massachusetts

Contract No. F19628-81-C-0001

REVIEW AND APPROVAL

This technical report has been reviewed and is approved for publication.

JUSTUS F. KENNEY, GS-14
Asst. DPD-Army Projects, JTIDS JPO

TONY L. HALL, Captain, USAF
Asst Chief Sys Spt Div, JTIDS JPO

6 Nov 81

HAROLD BINDER, GS-12
JTIDS Security Officer

WILLIAM S. JONES, COL
Deputy Program Director, Army
JTIDS JPO

FOR THE COMMANDER

NORMAN E. WELLS, Colonel, USAF
Program Director, JTIDS JPO

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER ESD-TR-81-388	2. GOVT ACCESSION NO. AD-A110 218	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) AN ALGORITHM SCHEMA FOR MINIMIZING BROADCAST TRANSMISSIONS IN PACKET RADIO NETWORKS		5. TYPE OF REPORT & PERIOD COVERED 6. PERFORMING ORG. REPORT NUMBER MTR-8384
7. AUTHOR(s) Bahaa W. Fam		8. CONTRACT OR GRANT NUMBER(s) F19628-81-C-0001
9. PERFORMING ORGANIZATION NAME AND ADDRESS The MITRE Corporation P.O. Box 208 Bedford, MA 01730		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Project No. 8280
11. CONTROLLING OFFICE NAME AND ADDRESS Deputy for Communications and Information Systems Electronic Systems Division, AFSC Hanscom Air Force Base, MA 01731		12. REPORT DATE DECEMBER 1981 13. NUMBER OF PAGES 48
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

ALGORITHMS
GRAPH THEORY
NETWORK CONTROL
NETWORKS
PACKET RADIO

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

In packet radio networks the broadcasting subnetwork required to reach a community of users can be substantially reduced through the algorithm schema presented. The algorithms given minimize the number of relay transmissions required to reach a given group of users in a broadcast network structure. The subnetwork produced by the algorithm schema also has the characteristic of minimizing the relay data to any user in the group. One algorithm gives a solution to the minimal cover problem (optimal except for a specific class of counterexamples) in polynomial time. (over)

DD FORM 1 JAN 73 1473

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. (Concluded)

Existing optimal algorithms are exponentially bounded in complexity.

ACKNOWLEDGEMENT

The author is indebted to W.C. Lee and I.D. Ross for their careful review of this document, and R.R. Thompstone for his help in the preparation of the diagrams contained in this document.

This report has been prepared by The MITRE Corporation under Project #8280. The contract is sponsored by the Electronic Systems Division, Air Force Systems Command, Hanscom Air Force Base, Massachusetts.

TABLE OF CONTENTS

PRECEDING PAGE BLANK-NOT FILMED

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION 1

INTRODUCTION

In time distributed multiple access (TDMA) communication networks it is often desirable to reach a large community of users through the broadcast transmission of information. Due to the TDMA architecture, those units participating in the transmission or reception of information packets are restricted from participating in other communication activities (on other channels) during that period. As a result of this the utilization of many relay units in a broadcast tree wastes a considerable quantity of system capacity. It would therefore be desirable to reduce, or minimize, the number of relay units used in a broadcast network. From this we can define the following problem to be considered:

Problem: Minimize the number of broadcast transmissions required to disseminate a packet of information to a community of users on a multi-relay TDMA network so that the relay delay to reach any user in the community is minimal.

The algorithm schema presented in this paper will solve this problem.

In addition to minimizing the total number of relay transmissions required to reach some community of users, it is also important that the information be disseminated in a timely manner. Thus it is desirable to minimize the delay, therefore to minimize the number of relay transmissions required to reach any member in the community. It is this objective that stipulates that the shortest path (with respect to relay "hops") from the originator to every member of the user community be known in advance. The algorithms presented are intended to be processed by a network control facility which has full knowledge of system connectivity.

Two algorithms are utilized in the schema where algorithm β is embedded in algorithm α.

The definitions needed to support a more formal representation of the problem will be given in the next section. The remainder of this paper will consider the problem and solution techniques in graph (set) theoretic terms with analogies to the communication problem previously defined.

SECTION 2

PROBLEM DEFINITION

In this section we will present a formal (mathematical) definition of the communication problem given in the introduction to this paper.

A number of other definitions will be required for the statement of the formal problem.

Definition 1: A graph G (V, E) is an ordered pair of disjoint sets V, E such that $E \subseteq V^2$ and $V \neq \emptyset$, where V is a set of vertices and E a set of edges.[1] [Bollobas, Pg xiii]

An edge, which is an element of set E, is denoted by its endpoints; thus if an edge connects vertices x and y it would be denoted as edge $\overline{xy}$. We can also state then that edge $\overline{xy}$ is adjacent to vertex x and to vertex y and if $\bar{xy} \in E$ we can then state that x and y are adjacent vertices. Additionally we may stipulate that an edge be directed, hence giving an orientation to the vertices that constitute its endpoints. A directed edge, $\overrightarrow{xy}$, would have a source x and a terminus y. This implies that if G (V,E) represents a network, then $\overrightarrow{xy}$ is a branch in the network along which a flow may pass from x to y but not from y to x. A directed edge $\overrightarrow{xy}$ in a graphical representation of a communication network would imply that y could hear x's transmissions but x could not receive messages transmitted by y.

Definition 2: A subgraph G' (V', E') of some graph G (V, E) is a graph in which $V' \subseteq V$ and $E' \subseteq E$.

From this definition it is clear that every graph G (V, E) has at least one subgraph.

Definition 3: A graph G(V,E) is said to be connected if for every $v_i, v_j \in V$ $(i \neq j)$ there exists at least one path(set of adjacent vertices) from v_i to v_j.

<u>Definition 4</u>: A spanning subgraph $\tilde{G}(\tilde{V}, \tilde{E})$, on a vertex set $V^{\circ} \subseteq V$, of $G(V, E)$ is a graph such that $V^{\circ} \subseteq \tilde{V}$ and $\tilde{E} \subseteq E$, where $\tilde{G}(\tilde{V}, \tilde{E})$ is connected.

Recall that the cardinality of a finite set of distinct elements S is the measure of the discrete elements in the set. Thus if some set S contained n discrete elements then cardinality $\left[S \right] = n$.

<u>Definition 5:</u> A minimal spanning subgraph $\tilde{G}(\tilde{V},\tilde{E})$ on a vertex set $V^{\circ} \subseteq V$ is a subgraph of $G(V, E)$ where $V^{\circ} \subseteq \tilde{V}$ and $\tilde{E} \subseteq E$ such that $\tilde{G}(\tilde{V},\tilde{E})$ is connected and that there exists no $\hat{G}(\hat{V},\hat{E})$ where

$$\text{cardinality} \left[\hat{E} \right] < \text{cardinality} \left[\tilde{E} \right]$$

and where $\hat{G}(\hat{V},\hat{E})$ satisfies the same conditions on V°.

<u>Definition 6</u>: Given a vertex $v_i \in V$ of a graph $G(V,E)$ the <u>outdegree</u>, $d^{+}(v_i)$ is a measure of the number of directed edges $\overrightarrow{v_i v_j}$ $(j \neq i)$ in the set of edges E.

Definition 7: Given a vertex $v_i \in V$ of a graph $G(V,E)$, the indegree , $d^-(v_i)$, is a measure of the number of directed edges $\overrightarrow{v_j v_i}$ $(i \neq j)$ in the edge set E.

We will now define another operator of measure on the sets V,E defining a graph $G(V,E)$.

Definition 8: Let $\Omega\left[G(V,E)\right]$ represent the number of vertices $v_i \in V$ where $d^-(v_i) > 0$ and $d^+(v_i) > 0$. More formally:

$$\Omega\left[G(V,E)\right] = \text{card}\left[\left\{ v_i \in V \;\middle|\; d^+(v_i), d^-(v_i) > 0 \right\}\right]$$

The shortest path from any $v_i \in V$ to any $v_j \in V$ $(v_i \neq v_j)$ shall be denoted as P_{ij}. P_{ij} shall then be an ordered set of vertices $v_i, \ldots v_k, v_l \ldots . v_j$ where the existence of any two consecutive vertices $v_k v_l$ in P_{ij} implies that a directed edge $v_k v_l$ is contained in the shortest path P_{ij}.

Let $G(V,E)$ represent a communications network where V is the set of terminals in the network, then any edge $\overrightarrow{v_i v_j} \in E$ implies that v_j can receive transmissions from v_i.

Let S denote the source (originator) in a given broadcast network and define a subgraph $\hat{G}(\hat{V},\hat{E})$ as the subgraph showing the union of the shortest paths (defined by vertices) from S to all users in a community of interest, V^o and all adjacent edges to $\hat{V}$: $\hat{E}$.

$$\hat{G}(\hat{V},\hat{E})=G\left(\left\{v_i \in P_{sj} \quad \forall v_j \in V^o\right\},\left\{\overrightarrow{v_k v_l} \in P_{sj} \quad \forall v_j \in V^o\right\}\right)$$

We can state the communication problem presented in the previous section as:

Problem Φ_1 Find the minimal spanning subgraph $\tilde{G}(\tilde{V},\tilde{E})$ of the subgraph $\hat{G}(\hat{V},\hat{E})$ on the vertex set $\left\{ V^o \cup S \right\}$.

We now add the constraint that in $\tilde{G}(\tilde{V},\tilde{E})$ the path P_{sj} from s to any $v_j \in V^o$ must be such that

$$\text{cardinality}\left[\tilde{P}_{sj}\right] = \text{cardinality}\left[P_{sj}\right]$$

thus retaining the minimal distance from any $v_j \subseteq V^o$ to the source defined by the set of shortest paths $\left\{P_{sj}\ \forall v_j \in V_o\right\}$ given on $\hat{G}(\hat{V},\hat{E})$.

Finally we state the condition of optimality: $\tilde{G}(\tilde{V},\tilde{E})$ is such that $\nexists$ minimal spanning graph $\hat{G}(\hat{V},\hat{E})$ which satisfies the same conditions and where:

$$\Omega\left[\tilde{G}(\tilde{V},\tilde{E})\right] > \Omega\left[\hat{G}(\hat{V},\hat{E})\right]$$

The full problem is given below.

<u>Problem</u> Φ_2: Find $\tilde{G}(\tilde{V},\tilde{E})$ such that:

1. $V^o \subseteq \tilde{V}$
2. cardinality $\left[\tilde{P}_{sj}\right]$ = cardinality $\left[P_{sj}\right]$ $\forall\, v_j \in V^o$
   where P_{sj} is the shortest path from S to v_j defined on $\hat{G}(\hat{V},\hat{E})$
3. $\tilde{G}(\tilde{V},\tilde{E})$ is a minimal spanning subgraph of G(V,E) on the vertex set $\left|V^o \cup S\right|$
4. $\nexists\, \hat{G}(\hat{V},\hat{E})$ which satisfies the constraints 1,2,3 and is such that

$$\Omega\left[\hat{G}(\hat{V},\hat{E})\right] < \Omega\left[\tilde{G}(\tilde{V},\tilde{E})\right]$$

Note that if any v_j is deleted from some set V, then all edges v_jv_k $(v_k \neq v_j)$ are no longer contained in the set E. Thus if

$V = V - v_j$ then $E' = E - v_jv_k$ $\forall v_j \neq v_k$.

Having defined the problem we can present the algorithm schema to solve it in the following sections.

SECTION 3

ALGORITHM α

This algorithm is designed to solve the problem ϕ_2 stated in the previous section. If not for the constraint:

$$\text{card}\left[\tilde{P}_{sj}\right] = \text{card}\left[P_{sj}\right] \ \forall_j \big| v_j \in V^o$$

an existing class of algorithms could be applied on $\hat{G}$ ($\hat{V}$ $\hat{E}$) to obtain a minimal spanning tree. Algorithm α preserves the shortest path characteristics on $\hat{G}$ ($\hat{V}$,$\hat{E}$) hence is not contained in the general class of minimal spanning tree algorithms.

This algorithm (algorithm α) utilizes another algorithm within its structure, which will be defined in the next section of this paper.

What algorithm β does is to find the minimal vertex cover on a bipartite graph. For the present we will only present one definition to explain the result of algorithm β when applied to some graph.

Definition A bipartite graph G (Y, X) is a graph where any $v_i v_j \in E$ is such that $v_i \in X$ and $v_j \in Y$ and $X \cap Y = \emptyset$, $X \cup Y = V$

Algorithm β when applied to a bipartite vertex set (Y,X) finds $X^\Phi \subseteq X$ where X^Φ is such that there is at least one vertex in X^Φ adjacent to any vertex in Y. Thus if X^Φ is such that $\nexists$ X" where

$$\text{card}\ [X''] < \text{card}\ [X^\Phi]$$

where X" satisfies the same conditions, X^Φ is called minimal cover of Y.

We say algorithm β finds a reduced cover X^{ϕ} because in its present form there exists a small class of examples where X^{ϕ} is not a minimal cover. Algorithm β is therefore not optimal by definition.

In the formulation of algorithm α that will be presented, algorithm β will be utilized in the form of an operator. The step

$$X_{i-1} = \text{algo}\ \beta \left[Y,\ X_i \right]$$

indicates then that X_{i-1} is the reduced cover of the bipartite graph $G(Y, X_i)$ generated by algorithm β.

We need the following definition.

<u>Definition:</u> $\Gamma(v_i)$ is the set of vertices adjacent to some vertex v_i. $\Gamma(v_i, X)$ is the set of vertices in X adjacent to v_i. $\Gamma(Y,X)$ is the set of vertices in set X adjacent to the vertices in set Y.

The following information must be known prior to executing algorithm α.

1. Adjacency Matrix of the graph G (V, E)

2. The source S, the set V^o

3. The set of shortest paths $P = \left\{ P_{sj} \quad \forall v_j \middle| v_j \in V^o \right\}$

We must define two set operators, one of which will be utilized within the structure of algorithm α .

Operator ψ : If P_{ij} is an ordered set of vertices then $\psi(P_{ij}, v_k)$ is the position of v_k in the ordered set P_{ij}. Thus in the ordered set $P_{ij} = (v_i, A, B, C, v_j)$, $\psi(P_{ij}, A) = 2$

Operator T : If P_{sj} is the set of vertices defining a shortest path from s to vj on some graph G, and P is the set of sets : $\{P_{sj}, \forall\ v_j \in V^o\}$ then

$$T(P,n) = \left\{ \left\{ v_i \in P_{sj} \,\middle|\, \psi(P_{sj}, v_i) = n \right\} \forall P_{sj} \in P \right\}$$

The variables utilized in the structure of algorithm α are given below.

θ : The length of the longest path in P.

B_{i-1} This constitutes the set of vertices that are elements of V^o which have a shortest path from the source of length i-1.

A_i: This is th set of vertices at iteration (relay level) i which must be covered by the set $T(P, i-1) \cup B_{i-1}$.

Z_{i-1} This is the set of vertices not in the set B_{i-1} that will partially constitute the minimal cover of A_i.

T_i: The set of terminals who are to relay a message at level i (definition applies to communication example only).

R_{i-1}: The set of terminals who are to receive a message transmitted from the terminals in the set T_{i-1} (definition applies to communication example only).

Two formulations of algorithm α will be presented. The first algorithm α_1 will solve the communications problem stated in the introduction by specifying (using T_i, R_{i-i}) the terminals who are to receive and transmit at every relay level.

Algorithm α_2 is less complex in its structure and solves the problem ϕ_2 given in the previous section.

Prior to the formal presentation of the two algorithms, a less exact formulation of algorithm α_1 will be given.

Algorithm α_1

We wish to solve the communication problem stated in the first section. Our objective is to minimize the number of terminals that must transmit and receive at any relay level such that a packet of information transmitted from the source will reach a prespecified community of interest, V^o, with minimal delay.

0. Based upon knowledge of P, the set of shortest paths on the network, find the length of the longest of the shortest paths from the source to a terminal in the set V^o. Let θ equal this distance.

1. Set the iteration counter i to θ .

2. Compute the set A_θ as the set of terminals in V^o that have shortest paths to the source of length θ .

3. Determine the set B_{i-1} , the set of terminals in V^o whose shortest paths to the source are of length i-1.

4. Apply algorithm β to find the minimal cover of the set of terminals in A_i that are not covered by the set of terminals in B_{i-1}.

5 Find the minimal cover of the terminals in $\Gamma\left(B_{i-1}, A_i\right)$, the set of terminals in A_i that are covered by the set B_{i-1} (but not necessarily minimally covered). The minimal cover will be a subset of the set B_{i-1}. It is important to note that although every terminal in B_{i-1} must receive an information packet we wish to minimize the number of terminals that must retransmit it . This is why we apply algorithm β in this step.

6. Determine the union of the set of terminals that must transmit at relay level i which are in V^o and those that must transmit at this level and are not (these are the set that must complete the minimal cover of A_i). Call the set comprising this union T_i.

7. Determine the union of B_{i-1} and those terminals not in B_{i-1} that complete the minimal cover of A_i. This is the set of terminals that must receive a packet of information at relay level i-1. Call this set R_{i-1}.

8. Let the set A_{i-1} be defined as the set R_{i-1} and let this constitute the set of terminals that must be covered in the next iteration of the algorithm.

9. Decrement the counter (set i=i-1) and if i=2 after being decremented then <u>STOP</u> . Otherwise go to step 3 and continue.

Algorithm α_1

0. $\theta = \text{Max}\left[\text{card} \left[P_{sj}\right], \forall_j \;\middle|\; v_j \in V^o \right]$

1. $i = \theta$

2. $A_\theta = \left\{ v_j \;\middle|\; v_j \in V^o, \text{card} \left[P_{sj}\right] = \theta \right\}$

3. $B_{i-1} = \left\{ v_j \;\middle|\; v_j \in V^o, \text{card} \left[P_{sj}\right] = i-1 \right\}$

4. Algorithm $\beta\left[\left(A_i - \Gamma(B_{i-1}, A_i)\right), \left(\Upsilon(P, i-1) - B_{i-1} \right)\right] = Z_{i-1}$

5. Algorithm $\beta\left[\Gamma(B_{i-1}, A_i), B_{i-1}\right] = Y_{i-1}$

6. $T_i = Z_{i-1} \cup Y_{i-1}$

7. $R_{i-1} = Z_{i-1} \cup B_{i-1}$

8. $A_{i-1} = Z_{i-1} \cup B_{i-1}$

9. $i = i-1$, if $i > 2$ go to 3

Algorithm α_2

0. $\theta = \text{Max}\left[\ \text{card}\left[P_{sj}\right],\ \ \forall\ v_j \in V^o\ \right]$

1. $i = \theta$

2. $A_\theta = \left\{ v_j \mid v_j \in V^0, \text{card}\left[P_{sj}\right] = \theta \right\}$

3. $B_{i-1} = \left\{ v_j \mid v_j \in V^o, \text{card}\left[P_{sj}\right] = i-1 \right\}$

4. $\text{Algorithm}\,\beta\left[\left(Ai - \Gamma(B_{i-1}, A_i)\right), \left(\mathcal{T}(P, i-1) - B_{i-1}\right)\right] = Z_{i-1}$

5. $A_{i-1} = B_{i-1} \cup Z_{i-1}$

6. $i = i-1$, if $i > 2$ go to 3

The computational complexity of algorithms α_1, α_2 is of order (V^4) based on the argument that algorithm β has complexity $O(V^3)$. More exactly the complexity of algorithms α_1 and α_2 is

$$\theta \left[\text{Order(algorithm } \beta\text{)}\right]$$

where θ is the length of the longest shortest path in the network. In reality $\theta << V$, thus we can realistically say θ is relatively constant with respect to V, therefore for practical purposes we can say the complexity of algorithms α_1 and α_2 is of order(V^3).
Algorithm β is polynomially bounded thus showing an improvement over the exponentially bounded algorithms previously developed for solving minimal cover problems.

SECTION 4

ALGORITHM β

It was shown in the previous section that algorithm β, required to solve the minimal cover problem, is embedded in the structure of algorithm α. In the communication example presented, algorithm β would determine the minimal number of terminals at relay level i that would have to transmit a packet so that every terminal at level i + 1 would receive a transmission. The related problem in graph theory is known as the minimal vertex cover problem.

INTRODUCTION

The problem of finding the minimal cover set $X^{*} \subseteq X$ in a bipartite graph G (X,Y) has been characterized as Steiner's second problem ([2]HAKIMI 1). In the past a number of algorithms have been developed to solve this problem but none have achieved a polynomial bound. Previous attempts at solutions ([3]BERGE, [4]HAKIMI) have been based on Boolean techniques and require an exponential number of computations. A second approach, utilizing 0-1 integer programming has been attempted but with a similar theoretic bound and is only applicable to small problems.

In this paper we will present an algorithm to solve the reduced cover problem with a polynomial number of computations. The solution to this problem ([4]HAKIMI) is believed to hold the key to a variety of cover problems posed by Steiner and known as problems 3, 4, 5. The interrelationship between these problems has been shown by Hakimi.

Algorithm β can be claimed only to solve Steiner's second problem heuristically because in its present form it fails to find an optimal (minimal) cover for a small class of counterexamples. Even in the heuristic form it is of significance in the communications field as in all cases it will find a near optimal cover.

The algorithm will first be presented informally and will, later in this section, be given in a more exact form.

No definitions beyond those presented in the previous section will be necessary for either formulation of this algorithm.

ALGORITHM β

*Assume we wish to find a minimal cover :$X^{\Phi} \subseteq X$ for a set of vertices Y in a bipartite graph G (X,Y).

0. Initialize the working sets to be utilized in the algorithm.

$$\text{Set: } Y^* = Y$$
$$X^* = X$$
$$X^{\Phi} = \emptyset$$

1. Find which vertex in Y^* has the fewest adjacent vertices in X, let this be $\tilde{y}^*$. If more than one exists choose one at random.

2. Remove $\tilde{y}^*$ from Y^*

3. If $\tilde{y}^*$ has greater than one vertex adjacent to it, then go to step 3B else do 3A.

3A. - Let vertex that is adjacant to $\tilde{y}^*$, call this z, enter the minimal cover set X^{Φ}, remove it from X^*.

- Let the set of vertices $\Gamma(z)$ be removed from Y^*

- Go to step 4.

3B. This step is selected only if $\tilde{y}^*$ has greater than one vertex in X^* adjacent to it. In this case we apply a weighting scheme to determine which vertex in $\Gamma(\tilde{y}^*)$ should be selected for the minimal cover set.

Assign a weight to every $x_i \in \Gamma(\tilde{y}^*)$ according to the following scheme:

Let $x_i(c)$, the weight of $x_i \in \Gamma(\tilde{y}^*)$ be equal to the sum (for all $y_j \in \Gamma(x_i)$) of:

$$\left(\text{Card} \left[X^* \right] - \text{card} \left[\Gamma \left(y_j \right) \right] \right) .$$

- Thus x_i is given a value according to how many vertices in Y* it is adjacent to and inversly proportional to the number of vertices in X* each of these are adjacent to.

- Select the vertex in $\Gamma(\tilde{y}^*)$ with the maximal weight. If more than one of the vertices have the same maximal weight, select one at random. Let this vertex be z.

- Go to 3A.

4. If $Y^* = \emptyset$ then all vertices are covered with X^ϕ STOP, else go to 1.

$*X^\phi$ is a minimal cover of Y

As can be seen by the description of algorithm β given above this technique relies on a weighting scheme. It has a worst case complexity V^3 if the degree of connectivity of each vertex is computed in advance and updated in each step; V^3 preprocessing steps required in advance to compute degree of connectivity, V steps required to update at each iteration. The total complexity of algorithm β is $\approx V^3$ where

$$V = \max \left[\text{card} \left[X \right] , \text{card} \left[Y \right] \right]$$

therefore complexity of the algorithm is:

$$0\ (V^3).$$

We now present the formal definition of algorithm β

Algorithm β

0. $Y^* = Y$, $X^* = X$, $X^O = \theta$

1. $\tilde{y}^* = \text{any } y_j \in Y^* \;\Big|\; \text{card}\left[\Gamma(y_j)\right] \leq \text{card}\left[\Gamma(y_k)\right] \quad \forall j \neq k$

2. $Y^* = Y^* - \tilde{y}^*$

3. If $\text{Card}\left[\Gamma(y^*)\right] > 1$, go to 3B, else let $z = \Gamma(\tilde{y}^*)$, go to 3A

3A. $Y^* = Y^* - \left[\Gamma\left(\Gamma(\tilde{y}^*)\right)\right]$, $X^* = X^* - \Gamma(\tilde{y}^*)$, $X^O\ X^O + z$ go to 4

3B. $\forall x_i \in \Gamma(y^*)$,

$$x_i(c) = \sum_{y_j \in \Gamma(x_i)} \left(\text{card}\left[X^*\right] - \text{card}\left[\Gamma(y_j)\right] \right)$$

$$z = x_i \;\Big|\; x_i(c) \geqq x_j(c) \quad \forall i \neq j, \left(x_i, x_j\right) \in \Gamma(y^*)$$

if $\text{card}\left[z\right] > 1$ select one element from z and let that element comprise z

1 Go to 3A

4. if $Y^* = \theta$ <u>STOP</u>, (X^O = solution set), else go to 2

SECTION 5

EXAMPLE UTILIZING ALGORITHM α_1

Presented in this section is an example to demonstrate the application of algorithm α_1. This will solve the communication problem stated in Section 1.

The network presented in this example was generated randomly.

The objective of algorithm α_1 when applied to the network was to minimize the number of terminals transmitting an information packet such that every unit in a community of interest receives the packet in minimal time.

The network contains 50 units, 23 of which are members of a prespecified community of interest.

Call this set V^o. If we label each terminal in the network with a number from 2 to 50, our community of interest can be defined as the following set:

$V^o = \left[8, 15, 17, 19, 21, 22, 23, 25, 27, 28, 30, 31, 32, 33, 38, 39, 40, 41, 44, 45, 47, 48, 49\right]$

S is the source terminal from where the transmissions originate that the terminals in V^o wish to monitor.

Figure 1 shows the example network where an edge between two vertices represents a bidirectional communications link.

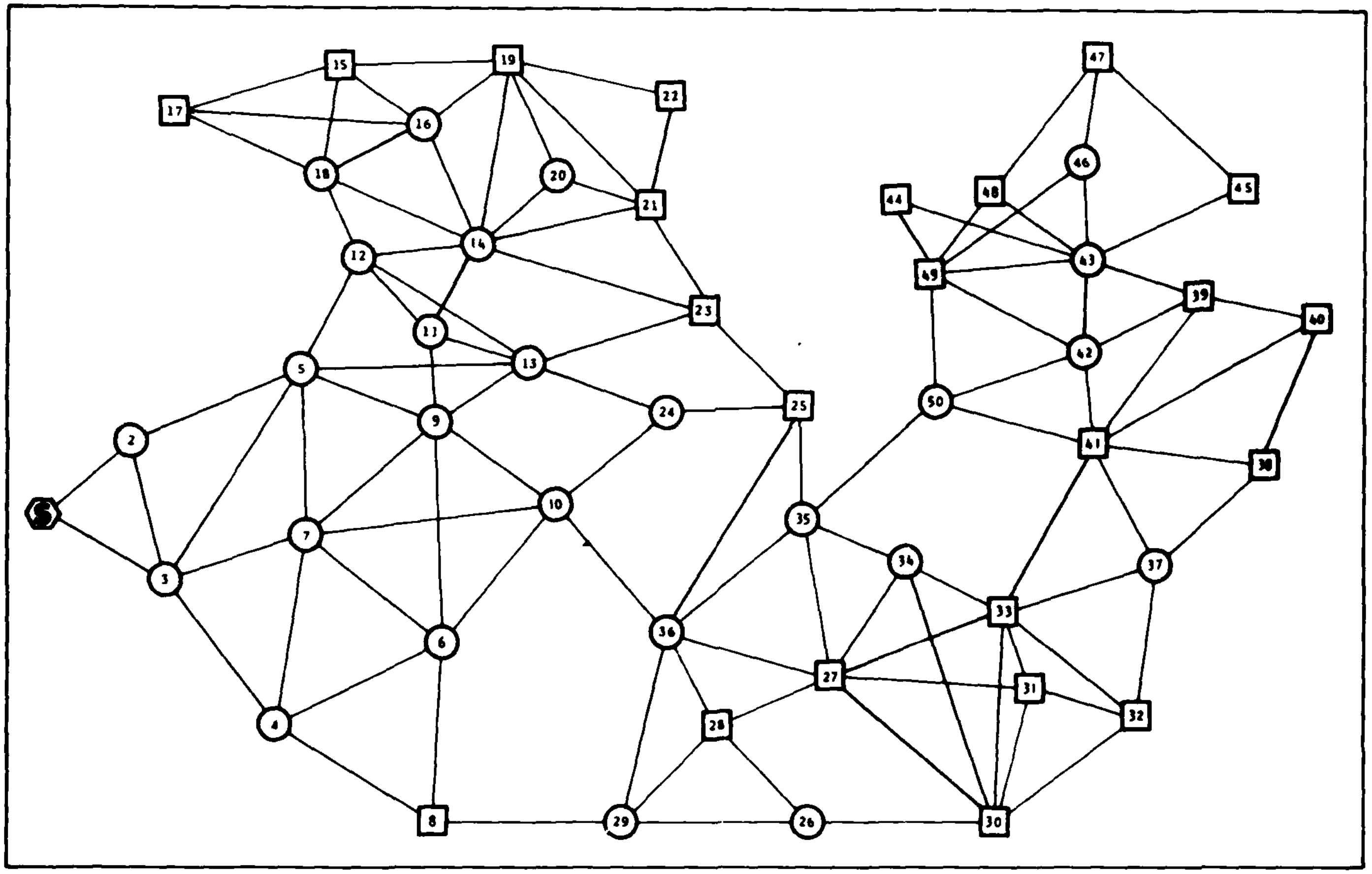


Figure 1

Example Network

Figure 2 shows the subgraph induced by the shortest paths from the source to all other terminals in the network. These paths were computed using Dijkstra's[5] algorithm, an algorithm commonly used in this application.

Below, algorithm α_1 is applied to the network to produce a minimal broadcast structure.

Algorithm α_1 is shown in each step of its structure through several iterations while algorithm β is stepped through once.

<u>Iteration 1</u> (algorithm α_1)

0. $\theta = 9$
1. $i = 9$
2. $A_9 = [45,\ 47]$
3. $B_8 = [38,\ 39,\ 40,\ 44,\ 48]$ (This set defines the vertices in V^o, the community of interest which are nine relay levels away from the source on their respective shortest paths).
4. Algorithm $\beta\left[A_9 - \Gamma(B_8,\ A_9),\ T(P,\ 8) - B_8\right] = Z_8$

$$\begin{cases} A_9 - \Gamma(B_8,\ A_9) = [45,\ 47] - [47] = [45] \\ T(P,\ 8) - B_8 = [38,\ 39,\ 40,\ 43,\ 44,\ 46,\ 48] - [38,\ 39,\ 40,\ 44,\ 48] = [43,\ 46] \end{cases}$$

$$\text{Algorithm } \beta\left[[45],\ [43,\ 46]\right] = [43] = Z_8$$

(Z_8 is the set of units not in B_8 that must transmit to relay level 9 in order to reach all units in A_9)

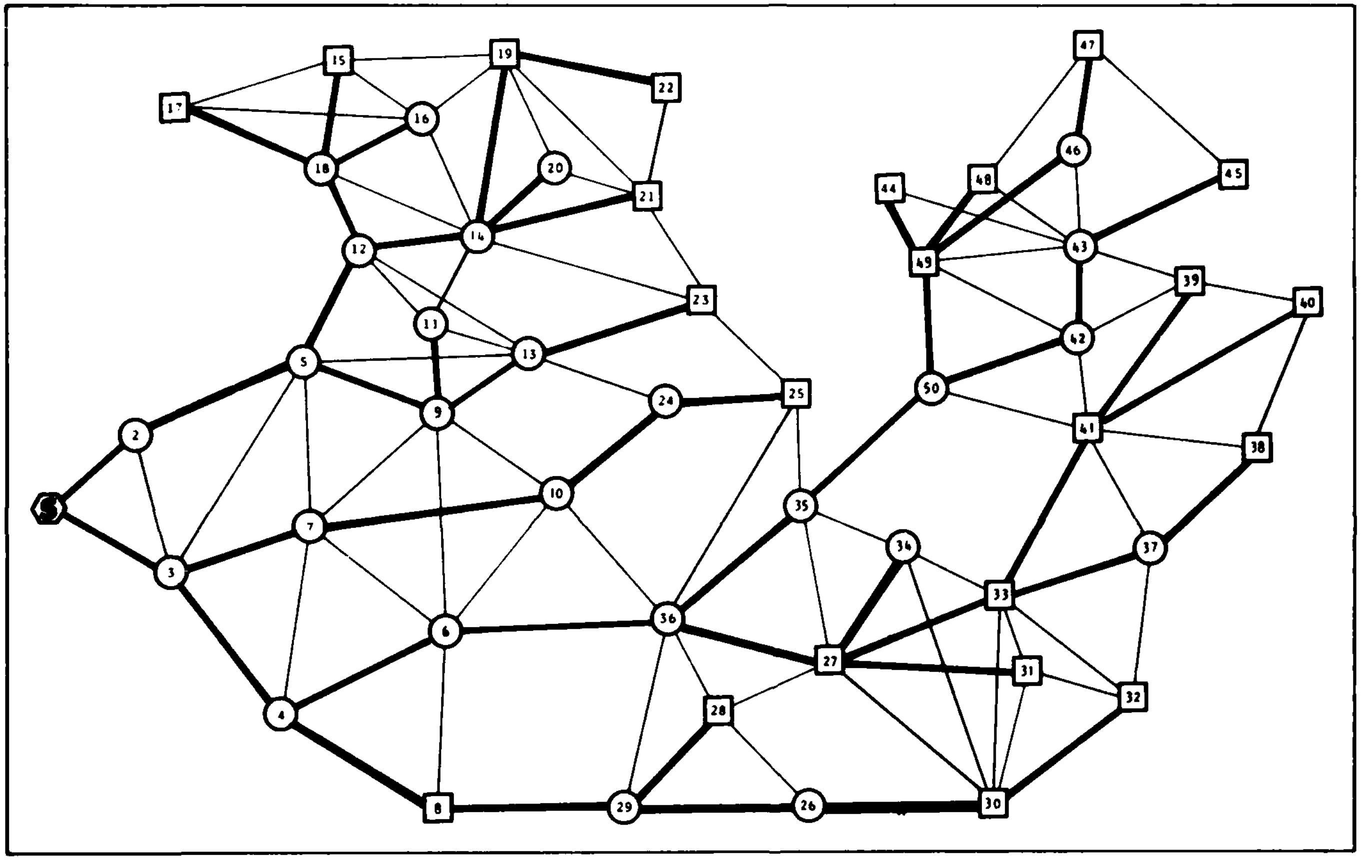


Figure 2

Broadcast Subnetwork Generated

by Dijkstra's Algorithm

5. Algorithm $\beta \left[\Gamma(B_8, A_9), B_8 \right] = Y_8$

$$\text{Algorithm } \beta \left[[47], [38, 39, 40, 44, 48] \right] = [48] = Y_8$$

(Y_8 defines the set of terminals in B_8 that must transmit to relay level 9)

6. $T_9 = Y_8 \cup Z_8 = [43, 48]$ (this constitutes the set of terminals that must relay the information at level 8 for terminals at level 9 to receive)

7. $R_8 = B_8 \cup Z_8 = [38, 39, 40, 43, 44, 48]$ (this constitutes the set of terminals which must receive messages at relay level 8)

8. $A_8 = B_8 \cup Z_8 = [38, 39, 40, 43, 44, 48]$

9. $i = i-1 \quad i = 9-1 = 8, \ 8 > 1 \Rightarrow$ go to 3

This starts iteration 2. Figure 3 shows the communication links selected thus far by algorithm α_1.

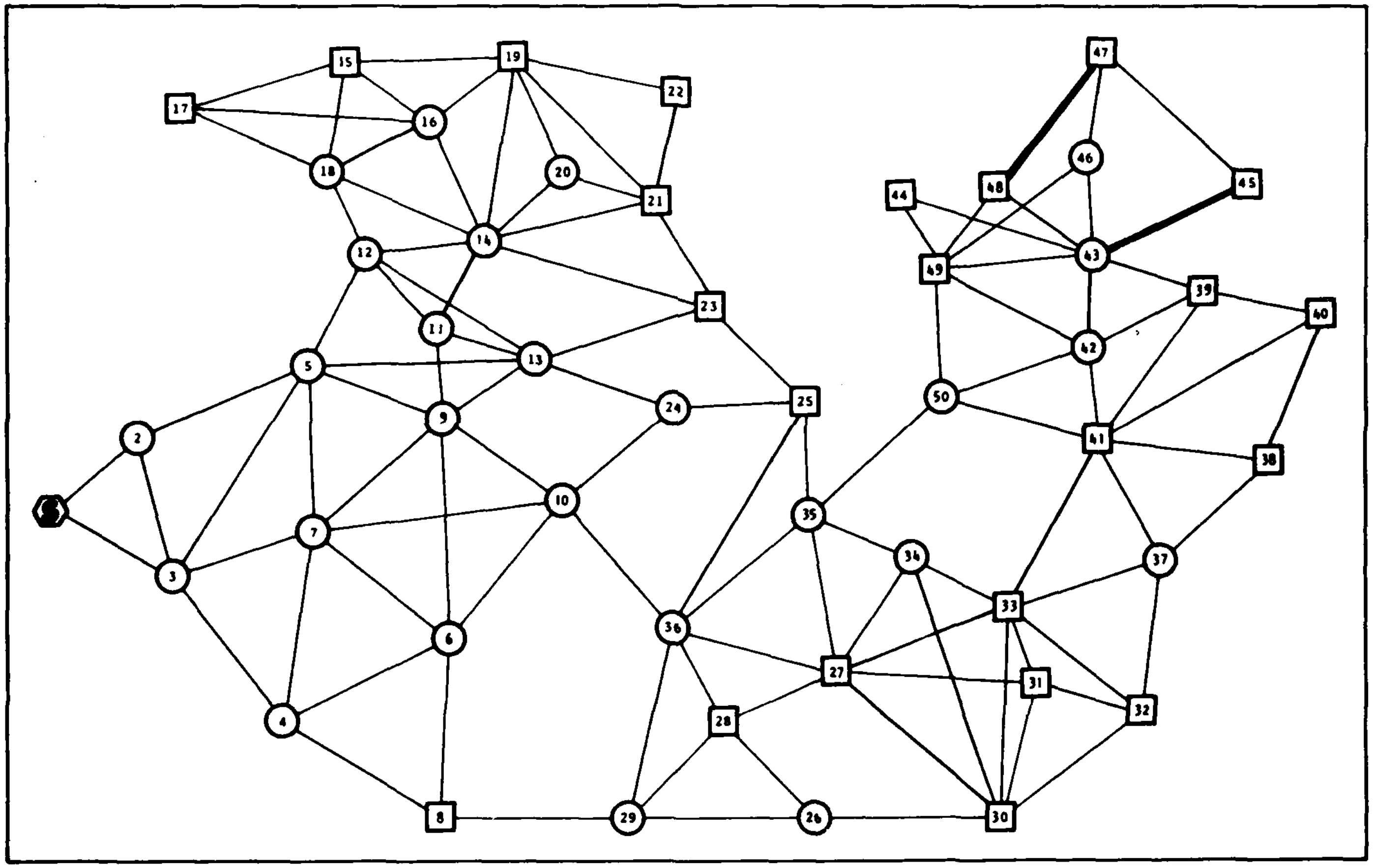


Figure 3

Subnetwork Produced After Iteration 1
of Algorithm α_1

<u>Iteration 2</u>

3. $B_7 = \left[32,\ 41,\ 49 \right]$

4. Algorithm $\beta \left[A_8 - \Gamma(B_7,\ A_8),\ \mathcal{T}(P,7) - B_7 \right] = Z_7$
   (in this case $\Gamma(B_7,\ A_8) = \left[38,\ 39,\ 40,\ 44,\ 48 \right] = A_8$)
   Algorithm $\beta \left[\emptyset,\ \mathcal{T}(P,7) - B_7 \right] = \emptyset = Z_7$

5. Algorithm $\beta \left[\Gamma(B_7,\ A_8),\ B_7 \right] = Y_7$
   Algorithm $\beta \left[\left[38,\ 39,\ 40,\ 44,\ 48 \right], \left[32,\ 41,\ 49 \right] \right]$
   $= \left[41,\ 49 \right] = Y_7$

6. $T_8 = \left[41,\ 49 \right] \cup \emptyset = \left[41,\ 49 \right]$

7. $R_7 = \left[32,\ 41,\ 49 \right]$

8. $A_7 = \left[32,\ 41,\ 49 \right]$

9. $i = 8-1$ go to 3

Figure 4 shows the links selected by algorithm α_1 in the first and second iterations.

Figure 5 shows the links selected in the first through third iterations of algorithm α_1.

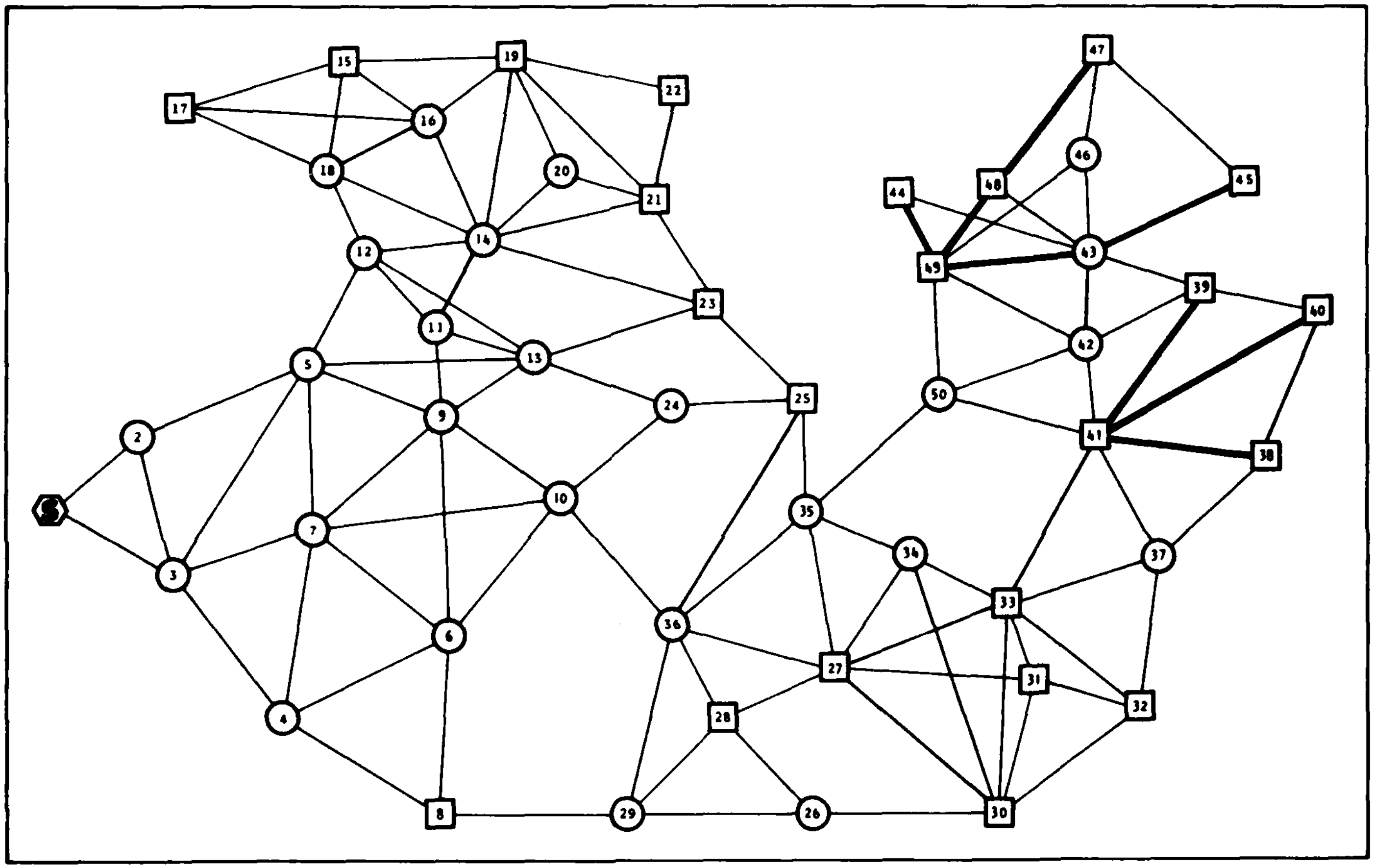


Figure 4
Subnetwork Produced After Iteration 2
of Algorithm α_1

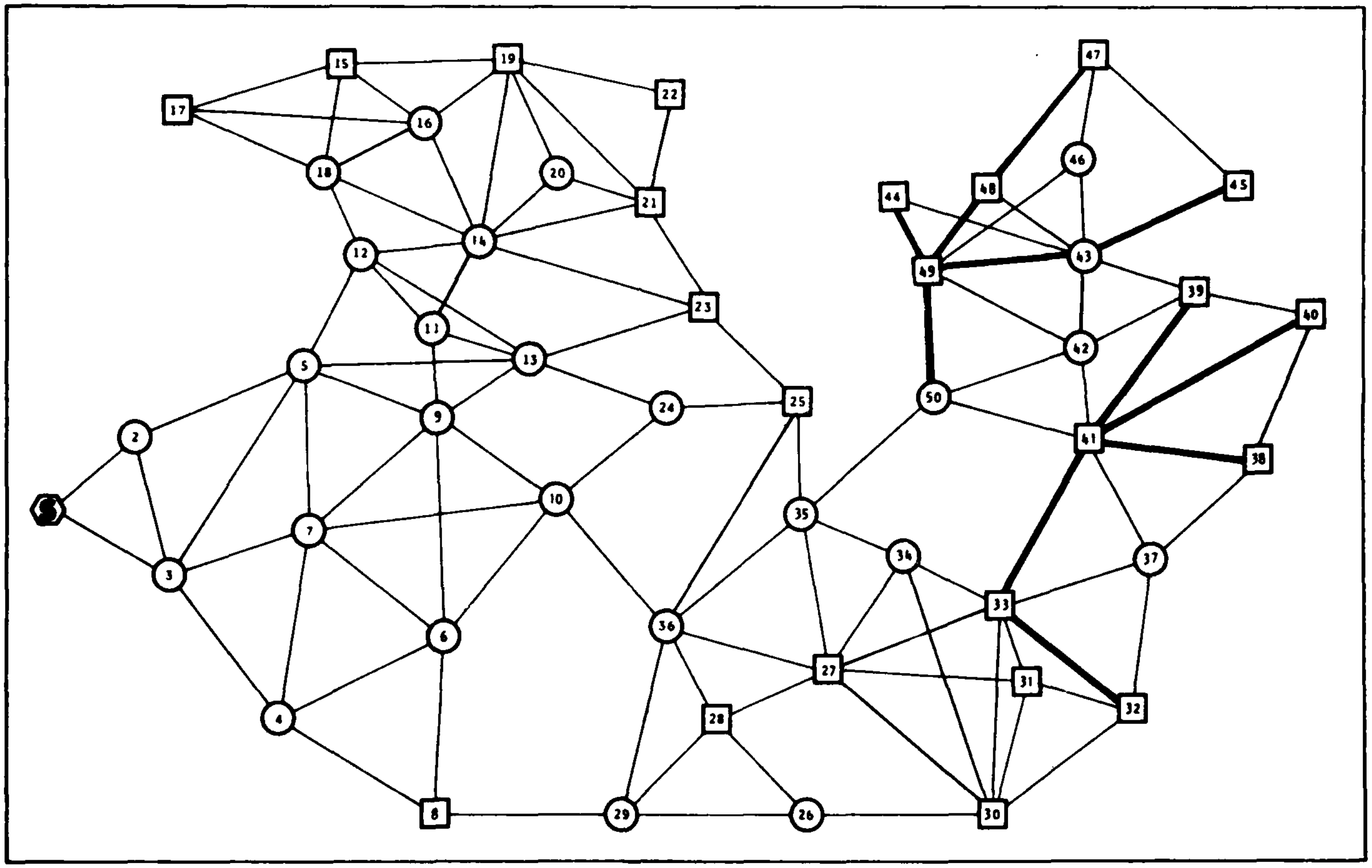


Figure 5

Subnetwork Produced After Iteration 3
of Algorithm α_1

Iteration 4 (algorithm α_1)

3. $B_5 = \left[15, 17, 19, 21, 23, 25, 27, 28 \right]$

4. Algorithm $\beta \left[A_6 - \Gamma(B_5, A_6), \Upsilon(P, 5) - B_5 \right] = z_5$

$$\begin{cases} A_6 = \left[22, 30, 31, 33, 50\right] \\ \Gamma(B_5, A_6) = \left[22, 23, 30, 31, 33\right] \\ \Upsilon(P,5) = \left[15, 16, 17, 19, 20, 21, 23, 25, 26, 27, 28, 35\right] \end{cases}$$

Algorithm $\beta \left[\left[50\right], \left[10, 20, 26, 35\right] \right] = \left[35\right] = z_5$

5. Algorithm $\beta \left[\Gamma(B_5, A_6), B_5 \right]$

Algorithm $\beta \left[\left[22, 30, 31, 33\right], \left[15, 17, 19, 21, 23, 25, 27, 28\right] \right]$

Algorithm β will be applied below to find the minimal cover:

$X^o \subseteq \left[15, 17, 19, 21, 23, 25, 27, 28\right]$ of the set $\left[22, 30, 31, 33\right]$

<u>Algorithm β</u>

0 $Y^* = 22, 30, 31, 33$

$X^* = 15, 17, 19, 21, 23, 25, 27, 28$

$X^o = \emptyset$

1 $\tilde{y}^* = 22$

2 $Y^* = Y^* - [22] = [30, 31, 33]$

3 card $[\Gamma(y^*)] = 1$ go to 3A

$\Gamma(y^*) = Z = [19]$

3A $Y^* = [22, 30, 31, 33] - [22] = [30, 31, 33]$

$X^* = [15, 17, 19, 21, 23, 25, 27, 28] - [19] =$

$[15, 17, 21, 23, 25, 27, 28]$

$X^o = [19]$

4 $Y^* = \emptyset$ go to 1

1 $y^* = [30]$

2 $Y^* = Y^* - [30] = [31, 33]$

3 $\Gamma(\tilde{Y}^*) = [27]$ card $[27] = 1$

$Z = [27]$ go to 3A

3A $Y^* = [30, 31, 33] - \Gamma(27) = [30, 31, 33] -$

$[30, 31, 33] = \emptyset$

$X^* = [15, 17, 21, 23, 25, 27, 28] - [27]$

$X^o = [19, 27]$

4 $Y^* = \emptyset$ STOP, X^o is a minimal cover

(end of algorithm β)

---

5. $\Rightarrow$ Algorithm $\beta \left[(B_5, A_6), \ B_5 \right] = \left[19, 27 \right] = Y_5$

6. $T_6 = \left[35\right] \cup \left[19, 27\right] = \left[19, 27, 35\right]$

7. $R_5 = \left[35\right] \cup \left[15, 17, 19, 23, 25, 27, 28\right]$

$$= \left[15, 17, 19, 23, 25, 27, 28, 35\right]$$

8. $A_5 = \left[15, 17, 19, 23, 25, 27, 28, 35\right]$

9. $i = 6-1 = 5 \Rightarrow$ go to 3

Figure 6 shows the links selected after the first through fourth iterations of algorithm α_1.

Figures 7, 8, 9, 10 show links selected through iterations 5, 6, 7, and 8 of algorithm α_1 .

Figure 11 shows the final minimal broadcast networks produced by algorithm α_1.

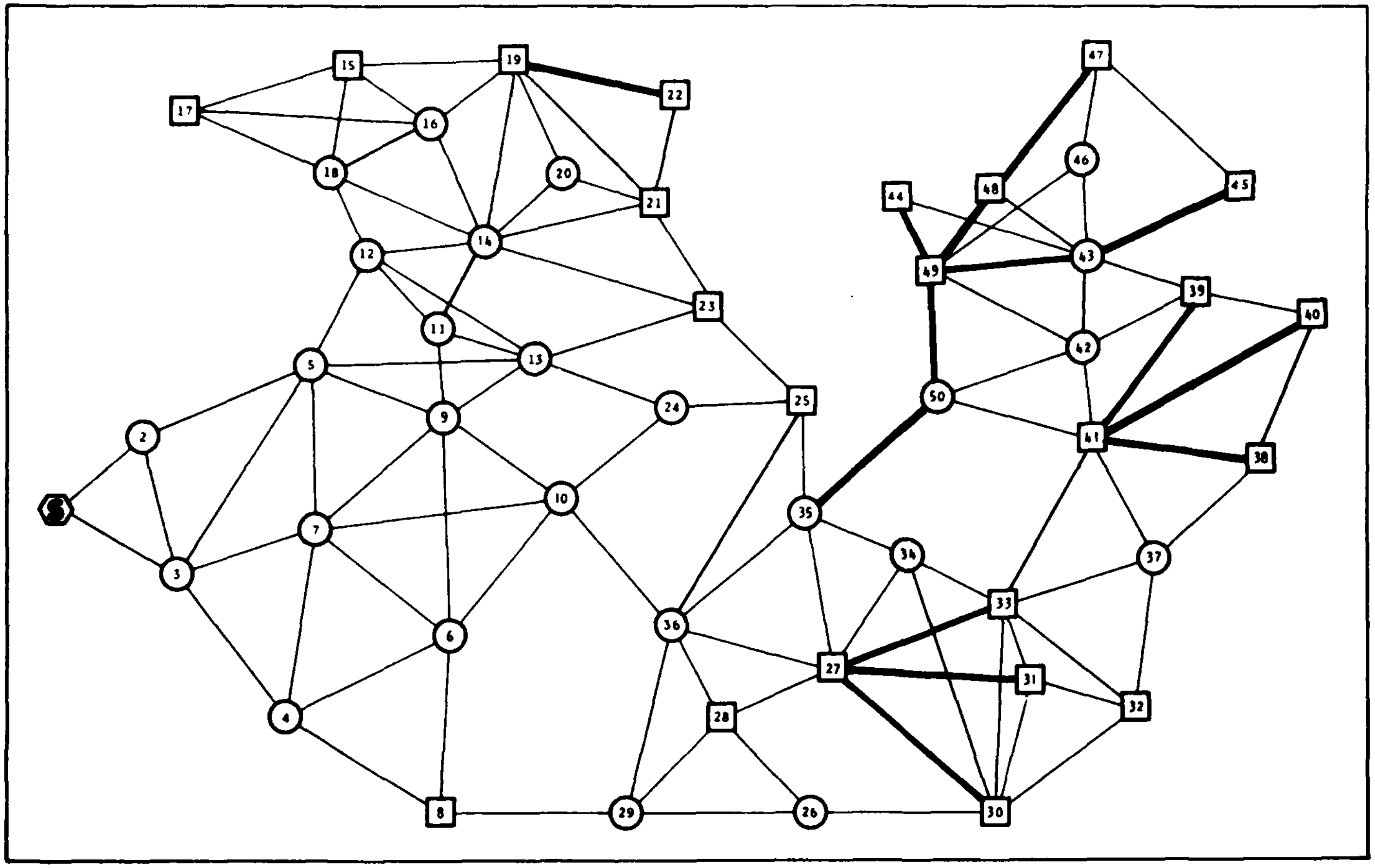


Figure 6

Subnetwork Produced After Iteration 4
of Algorithm α_1

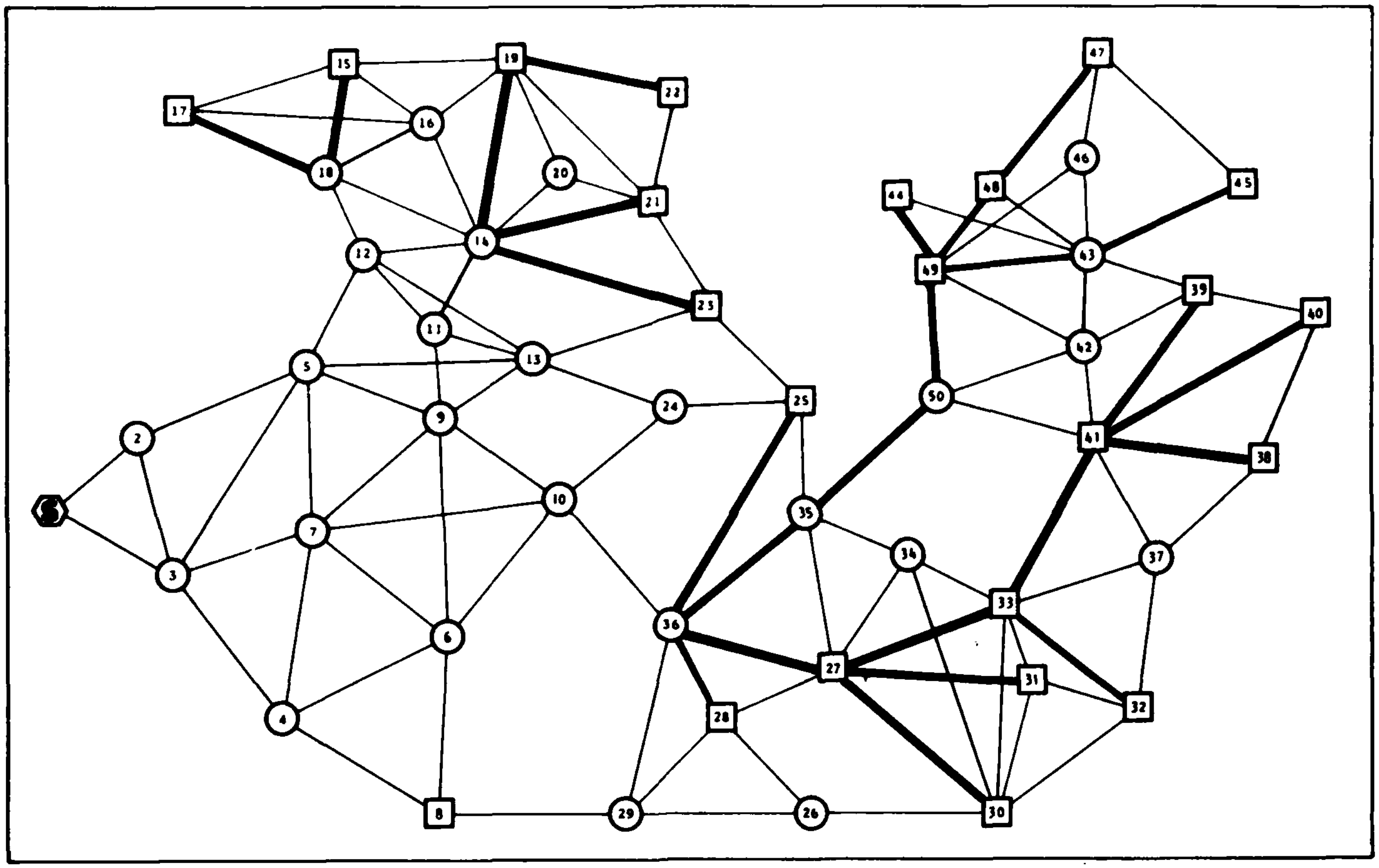


Figure 7

Subnetwork Produced After Iteration 5 of Algorithm α_1

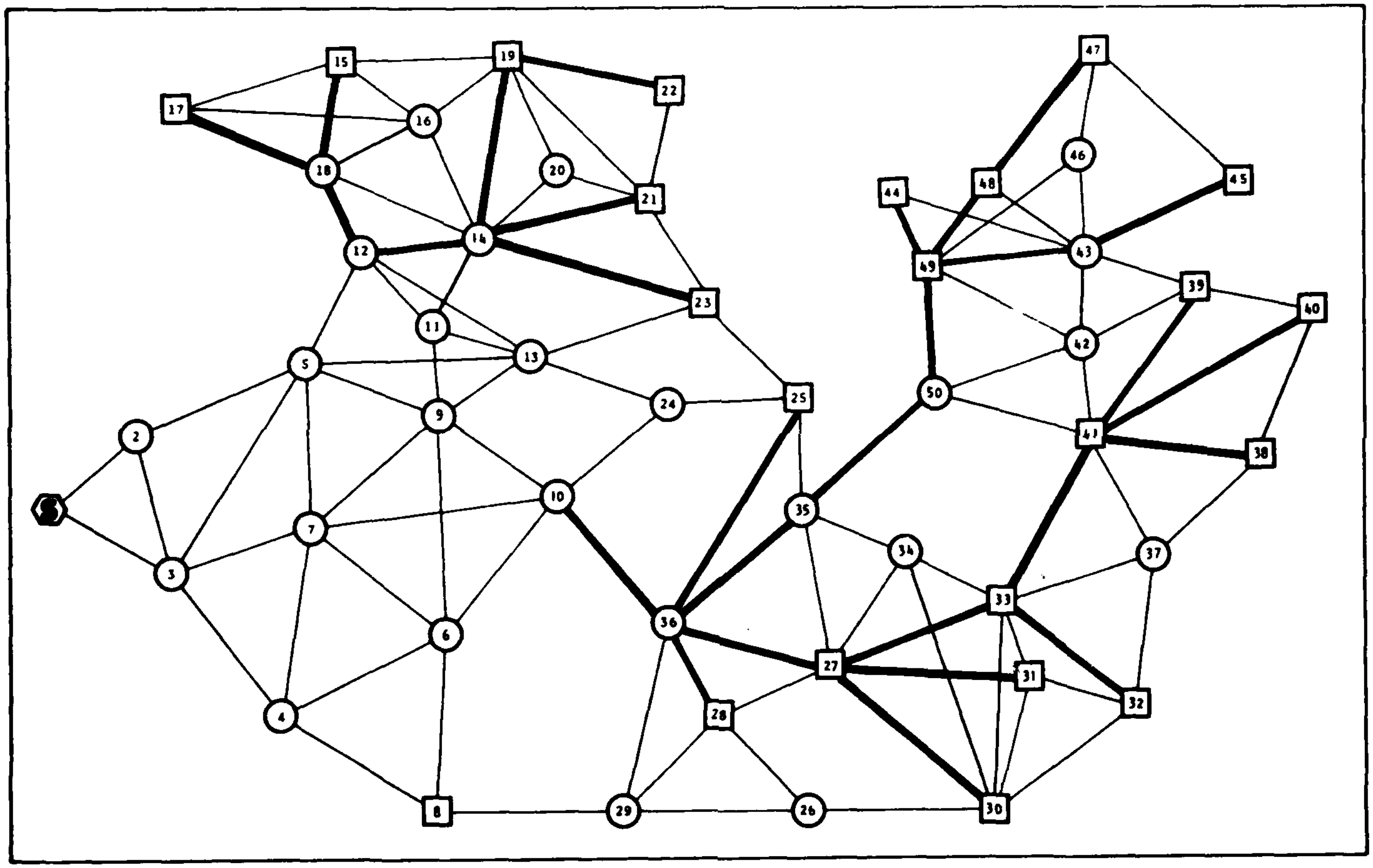


Figure 8
Subnetwork Produced After Iteration 6
of Algorithm α_1

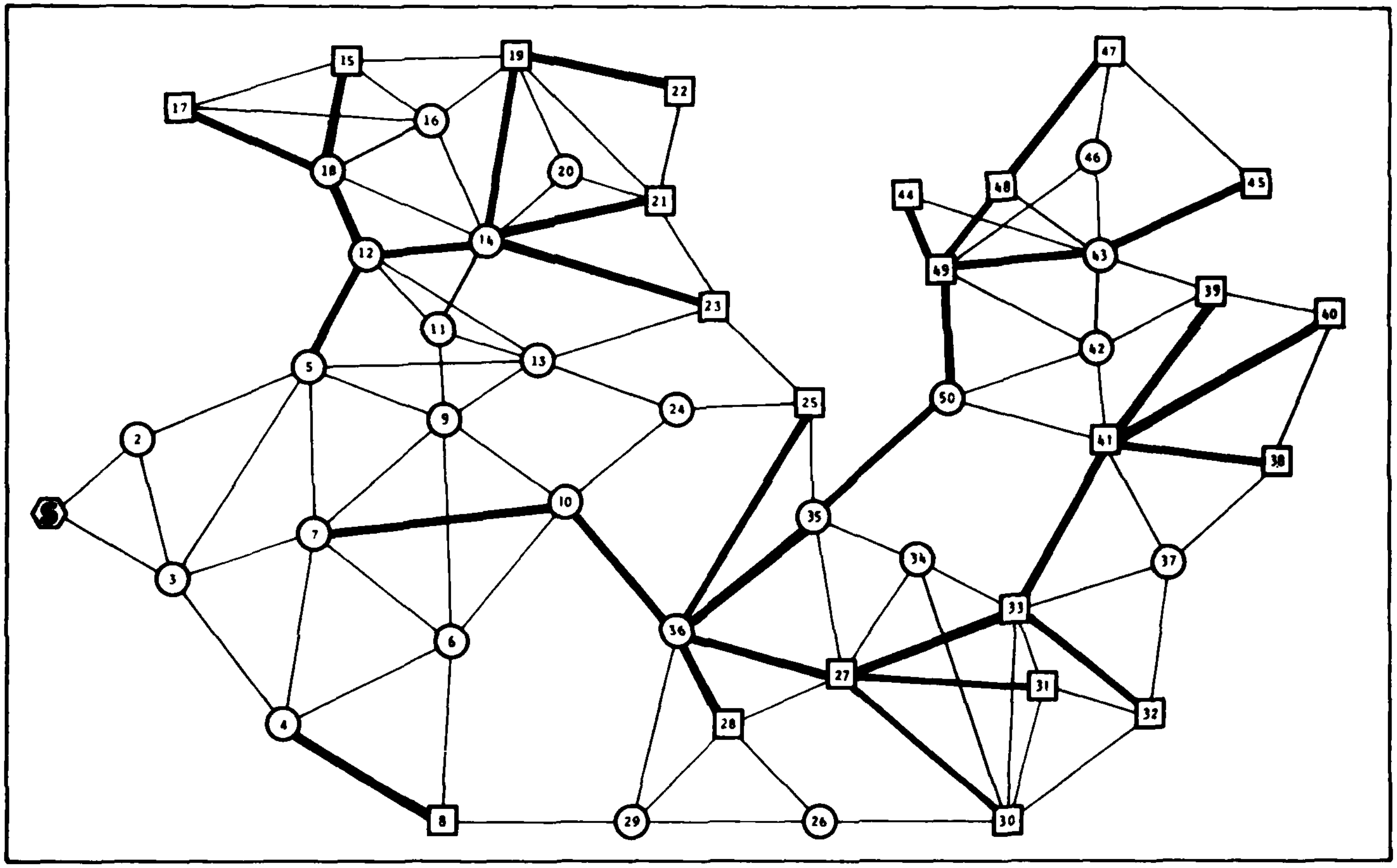


Figure 9
Subnetwork Produced After Iteration 7
of Algorithm α_1

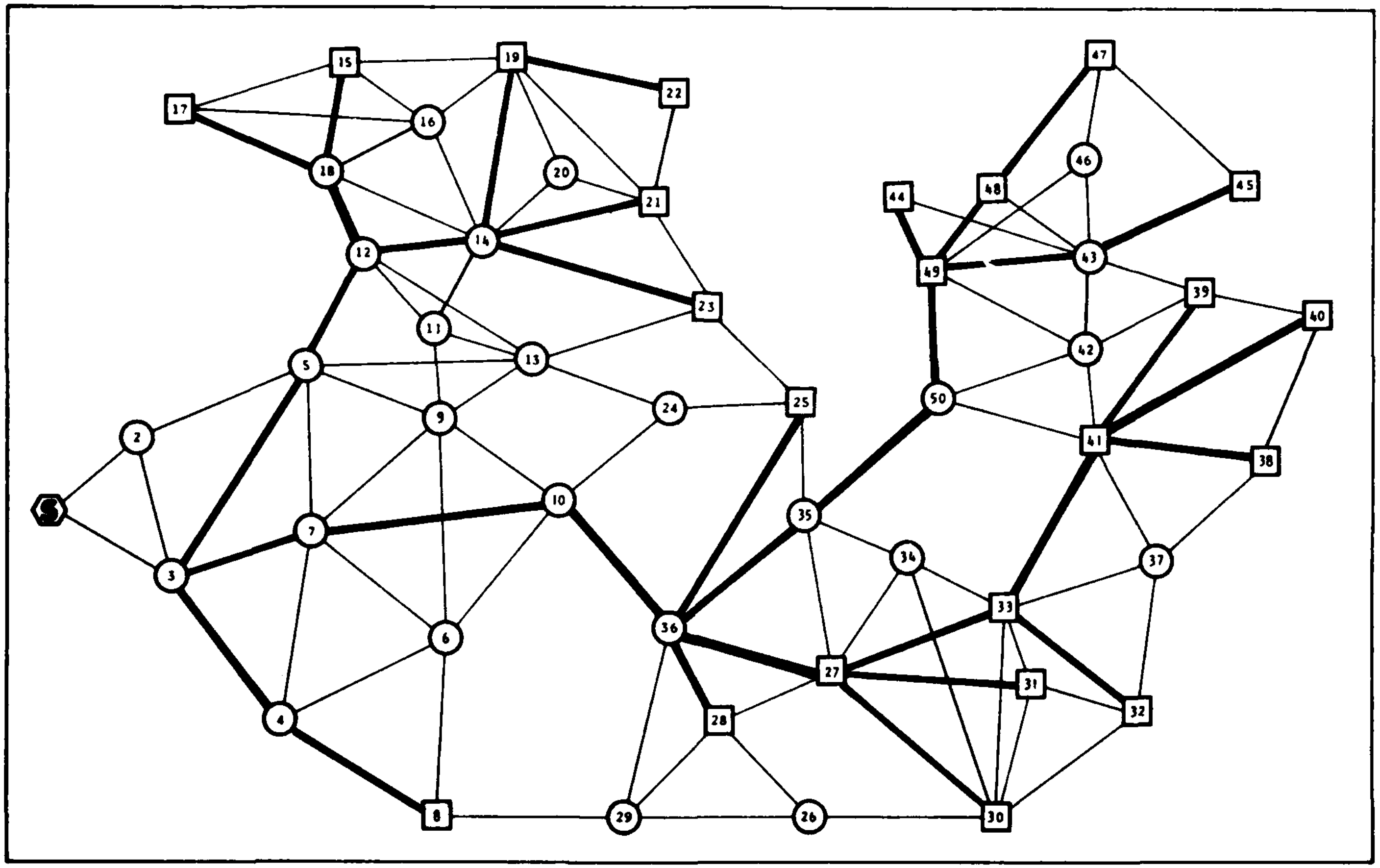


Figure 10

Subnetwork Produced After Iteration 8
of Algorithm α_1

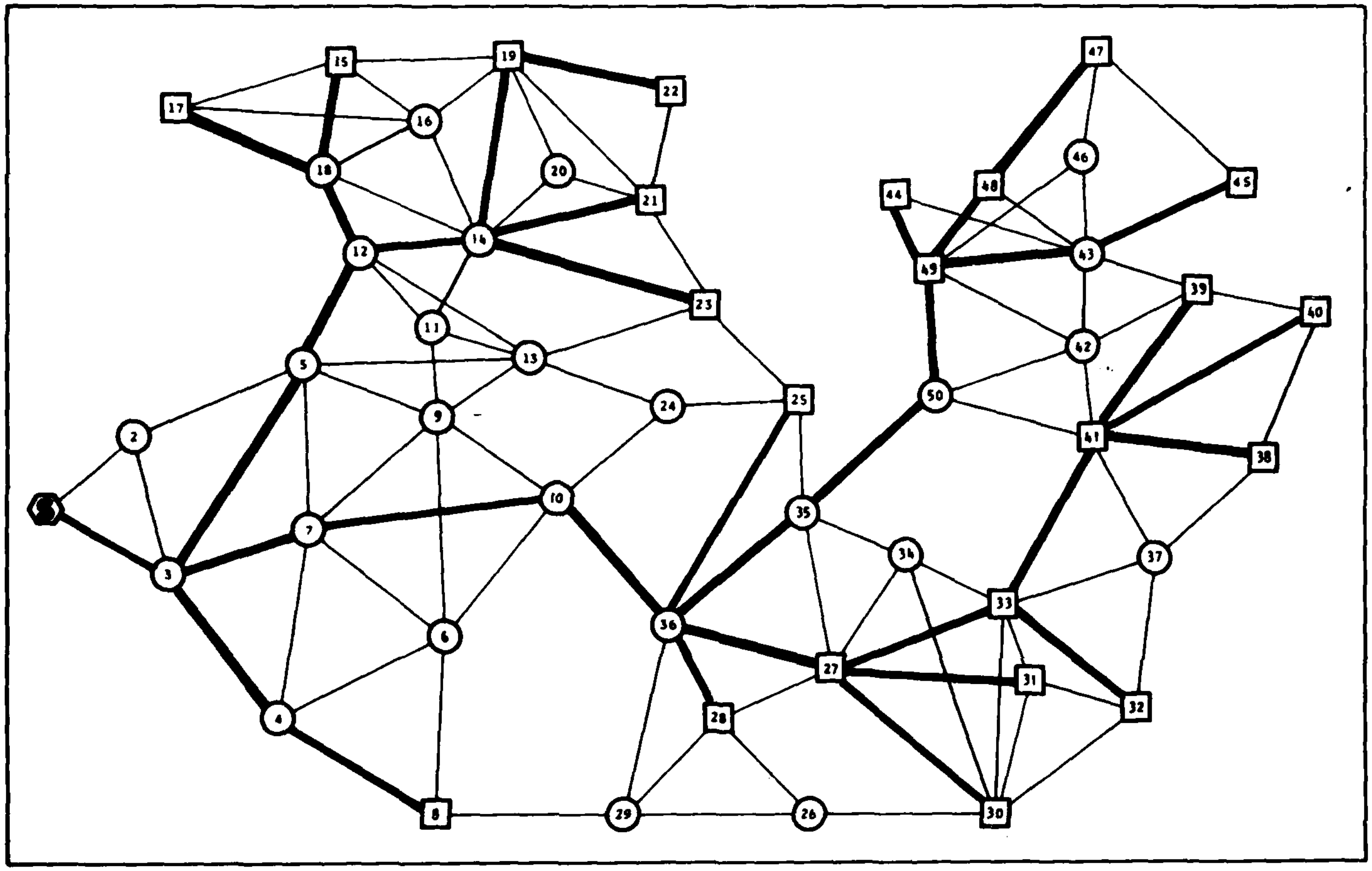


Figure 11

Reduced Broadcast Network

Generated by Algorithm α_1

Table 1 shows the transmissions by relay level in the broadcast network given by Dijkstra's algorithm compared to those in the broadcast network generated by algorithm α_1

TABLE 1

Performance of Algorithm α_1

Relay Level	Dijkstra		Algorithm α_1		% Difference	
	Trans.	Rec.	Trans.	Rec.	Trans.	Rec.
0	1	2	1	1	0	-50
1	2	3	1	3	-50	0
2	3	5	3	3	0	-40
3	5	6	2	3	-60	-50
4	6	11	3	8	-50	-27.3
5	5	6	3	5	-40	-16.7
6	2	5	2	3	0	-40
7	4	6	2	6	-50	0
8	2	2	2	2	0	0
Totals	30	46	19	33	-36.7	-28.3

It is clear in this example that algorithm α_1 produces a dramatically reduced broadcast network compared to that generated by Dijkstra's algorithm while maintaining the minimal delay characteristics of Dijkstra's results.

LIST OF REFERENCES

1. B. Bollobas, Extremal Graph Theory, Academic Press, London 1978

2. S. L. Hakimi (HAKIMI 1), "Steiner's Problem in Graphs and Its Implications", Networks Volume 1, 1971

3. C. Berge, The Theory of Graphs, Methuen and Co., London, 1962

4. S. L. Hakimi, "Optimum Distribution of Switching Centers in a Communication Network and Some Related Problems ", Operations Research Vol. 13, 1968, pp. 462-475

5. Bondy, U.S.R. Murty: Graph Theory With Applications, McMillan Press, London, 1976

DAT

FILM